

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 03-214235

(43)Date of publication of application : 19.09.1991

(51)Int.Cl.

G06F 9/38

(21)Application number : 02-008384

(71)Applicant : HITACHI LTD

(22)Date of filing : 19.01.1990

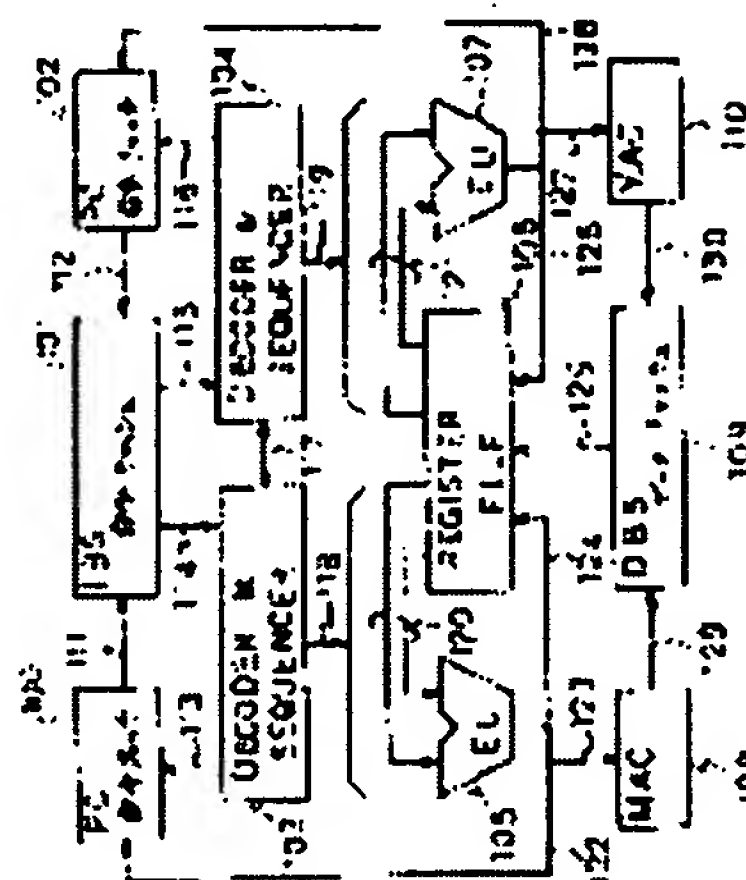
(72)Inventor : NAKATSUKA YASUHIRO  
KUROSAWA KENICHI

## (54) PARALLEL PROCESSING METHOD FOR PLURAL PATHS

(57)Abstract:

PURPOSE: To extract the 100% parallelism out of a relevant program by preparing plural instruction taking-out parts.

CONSTITUTION: When a breakup instruction is executed by a processing system, the instruction take-out means of other processing systems have accesses to an instruction cache memory 101 based on an instruction address set by a breakup instruction or an instruction address that is already set. Thus, the parallel processing can be carried on in each processing system despite an internal branching occurred in an instruction train during processing at each side. Meanwhile it is required again to select one of those processing systems that start their actions independently of each other. In such conditions, a merging instruction is executed by a single instruction executing means 105 (107). Thus, the actions of the instruction taking-out means of other processing systems are stopped. Then the operation of the means 105 (107) is also stopped. Consequently, the branching instructions themselves can also be executed in parallel with each other.



BEST AVAILABLE COPY

## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's

## ⑫ 公開特許公報(A)

平3-214235

⑬ Int. Cl.<sup>3</sup>

G 06 F 9/38

識別記号

3 7 0 A

庁内整理番号

7927-5B

⑭ 公開 平成3年(1991)9月19日

審査請求 未請求 請求項の数 3 (全8頁)

⑮ 発明の名称 複数バス並列処理方法

⑯ 特 願 平2-8384

⑰ 出 願 平2(1990)1月19日

⑱ 発 明 者 中 塚 康 弘 茨城県日立市久慈町4026番地 株式会社日立製作所日立研究所内

⑲ 発 明 者 黒 沢 憲 一 茨城県日立市久慈町4026番地 株式会社日立製作所日立研究所内

⑳ 出 願 人 株式会社日立製作所 東京都千代田区神田駿河台4丁目6番地

㉑ 代 理 人 弁理士 秋本 正実

## 明 細 書

## 1. 発明の名称

複数バス並列処理方法

## 2. 特許請求の範囲

1. メモリから命令を取り出す命令取り出し手段、  
該手段により取り出された命令のデコード手段、  
及び該手段によりデコードされた命令の実行手段より成るところの独立動作可能な処理系の複数個と、分裂命令及び融合命令とを設けるとともに、1つの処理系により処理中の命令列に同時処理可能な命令が存在するときには、当該処理系で分裂命令を実行させることにより他処理系による同時並列処理を開始し、複数の処理系による同時並列処理を1つの処理系の処理へまとめるときには、当該1つの処理系によって上記融合命令を実行させることにより上記1つの処理系以外の上記同時並列処理を行っていた処理系の動作を停止することを特徴とする複数バス並列処理方法。

2. 前記複数の処理系によって同時並列処理を行

う場合に、複数の処理系の各々にフラグを設け、さらに上記処理系の処理対象とする命令列の中に条件休止命令を挿入するとともに、同時並列処理の開始を指示する前記分裂命令によって各処理系の上記フラグをクリアし、各処理系が上記条件休止命令を実行したときには自処理系のフラグを+1更新しかつ他処理系のフラグを調べ、もし自処理系と他処理系のフラグの値が異なるときはその値が一致するまで自処理系の次の命令の処理を休止することを特徴とする請求項1記載の複数バス並列処理方法。

3. 前記融合命令が実行されたときに、該融合命令を実行した処理系で継続される処理の連続する命令の一部を、該連続する命令が分岐命令を含まないときに上記融合命令により動作を停止された処理系により同時並列処理することを特徴とする請求項1記載の複数バス並列処理方法。

## J. 発明の詳細な説明

## 〔産業上の利用分野〕

本発明は、複数の命令を同時処理するための複

数バス並列処理方法に関するものである。

#### 〔従来の技術〕

演算処理の高速化のために、複数の命令を同時実行するようにした処理装置の例として、特開昭63-49843号、米国特許4,766,566号、及びヨーロッパ特許87110751.2号に記載されたものがあり、これらの特許は同一内容のものを各国に対して出願したものである。この特許出願における主張は、近年スーパースカラという名前で呼ばれている並列化方式であり、アドレスが連続する2つの命令を同時に実行するものである。米国特許4,766,566号のFig. 2-7に記載されているように、同時に実行される2つの命令は単一の命令バッファに納められており、これらが同時に実行される。この方式では、分岐が生じたときに連続した命令の実行はできないから、一方の実行ユニットで分岐命令が実行されたときに命令バッファをキャンセルして分岐発生時の処理を行っている。

#### 〔発明が解決しようとする課題〕

上記従来技術には、分岐命令による本質的な並

列度の減少という問題があった。即ち、一方の実行ユニットの分岐命令によって、命令バッファがキャンセルされてしまい、このとき2つの実行ユニットが独立して動作できない。従って、命令の並列度の抽出時に最適化を行う範囲がエントリと分岐命令との間のみに限定されてしまう。ここでエントリとは分岐命令の直先あるいは分岐命令の直後の命令を指す。分岐命令は5命令に1度以上の割合で出現するといわれており、この状態では最適化はほとんど意味を持たない。2倍のハードウェア物量を投入した割に2命令同時実行による性能向上の効果は少ない、あるいは殆どないという結果となる。

本発明の目的は、この分岐命令による並列度の減少を解決し、分岐命令自身の並列化も可能とする複数バス並列処理方法を提供することにある。

#### 〔課題を解決するための手段〕

上記の目的を達成するために、本発明においては、パイプライン構成された命令取り出し手段、デコード手段及び命令実行手段から成るところの

互いに独立動作可能な処理系を複数個設け、さらに1つの処理系がその命令を実行したときに他の処理系の動作を開始及び停止させるための分裂命令及び融合命令と、命令の実行がどこまで進んだかを示すフラグと、該フラグを参照して各処理系による命令の実行を制御する条件休止命令とを設けた。

#### 〔作用〕

1つの処理系により分裂命令が実行されると、他の処理系の命令取り出し手段が上記分裂命令によってセットされる命令アドレス、あるいはすでにセットされている命令アドレスに従い命令用キャッシュメモリをアクセスする。これにより、それぞれの命令取り出し手段は異なるプログラムカウンタを持つため、基本的に独立に動作し、従ってそれぞれの側で処理中の命令列の内部での分岐があっても、各処理系における並列処理を継続できる。

一方、このようにして独立に動作し始めた各処理系を再び1本に絞り込む必要性もでてくる。こ

のときは1つの命令実行手段により融合命令を実行することにより、他の処理系の命令取り出し手段の動作を停止させる。これにともない命令実行手段も停止される。即ちパイプラインが絞り込まれる。融合命令によって停止したパイプラインは動作中のパイプラインによる分裂命令あるいは一定時間ごとに行われるタイマチェック機構によってのみ再開される。

2つの処理系の処理の間にデータの依存関係がある場合は、コンパイラが有する実行順序の情報に従って実行コードの中に実行順序の制御のための命令、即ち条件休止命令を埋め込む。複数の処理系の処理のパスにはそれぞれ同期用のフラグがあって、これらは分裂命令によってその値を増加させられる。そして、条件休止命令は各処理系のフラグの内容とそのときの自処理系の処理結果によって次の処理を得つか否かを判断し、これによって必要なデータが得られた後に当該処理が行われるように各処理系の処理の実行制御を行う。

#### 〔実施例〕

以下、本発明の一実施例を詳細に説明する。第1図は本発明を応用したプロセッサのブロック図である。命令フェッチユニット100及び102は、それぞれの内部に独立動作するプログラムカウンタを持ち、信号線111及び112を用いて命令用キャッシュメモリ101に対して命令フェッチ要求を出す。キャッシュメモリ101はこれらの片方を交互にあるいは両方を同時に受け付ける。対応するデータは同時に受け付けられた場合にはそのまま、交互に受け付けられた場合にはキャッシュメモリ101の内部にバッファリングされて、各サイクルごとに信号線114及び115へ出力される。それぞれが独立に動作可能なデコーダやシーケンサから成る制御ブロック103及び104は、キャッシュメモリ101からの命令をデコードした情報をもとに、信号線118及び119を用いて下記の命令実行部を並列に制御したり、分岐命令などの情報をもとに信号線113及び116とデータ線122及び128を用いて命令フェッチユニット100及び102をも制御する。制御ブロック103及び104は信号線117によって相互に接

し、信号線125を介してレジスタファイル106に対してデータの供給を行う。

第1図における左右のリソースは特殊な場合を除いて独立に動作可能であり、プログラムの並列性を抽出し同時に実行するのに適した構成となっている。極端な場合を考えると、左右で全く別のプログラムを実行させることもできる。一方、第1図において左右で共有されている命令用キャッシュメモリ101、レジスタファイル106及びデータ用キャッシュメモリ109の内部構造は、プロセッサの実現方法や使用目的によって多少異なる可能性がある。即ちキャッシュメモリ101、109の内部は2つのキャッシュメモリから構成されてもよいし、1つのマルチポートキャッシュメモリから構成されてもよい。また、フェッチ要求を交互に受け付けることができる1つのキャッシュメモリで構成し、命令取り出しユニットに格納された命令をバッファリングする手段を持ち、あたかも毎サイクルフェッチしたように見せかけるという方法もある。レジスタファイル106の内部は左右で完全

に共有されており、例えば両者の同期をとるときなどにこの信号が利用される。命令実行部は1つのレジスタファイル106と2つの実行ユニット105及び107から構成される。レジスタファイル106の一部と実行ユニット105は制御信号118によって、レジスタファイル106の一部と実行ユニット107は制御信号119によってそれぞれ制御される。制御信号118と119によって制御されるレジスタファイル106の各部分の間には共通部分があってもなくてもよい。実行ユニット105及び107はそれぞれ独立したソースデータバス120及び121を持ち、またそれぞれ演算結果をファイル106へ格納するためのターゲットデータバス124及び126を持っている。メモリアクセスは、実行ユニット105及び107が計算したメモリアドレスをメモリアクセスコントローラ108及び110へ渡し、これらコントローラ108及び110はデータ用キャッシュメモリ109に対してアクセス要求128及び130を出すことにより行われる。キャッシュメモリ109はこれらの信号を交互あるいは同時に受け付け、ロード/ストア専用バ

スに共有されたレジスタファイルでもよいし、全く独立した2つのレジスタファイルから構成されてもよい。無論部分的に共有する形態のものも考えられる。共有される部分には、左右に分配される独立したバス120、121をもつ必要がある。

次に、このようにして実現されたシステムの動作を第2図を用いて説明する。クロック信号K1はマシンサイクルの前半で高レベル、後半で低レベルの信号であり、この1サイクルで各部が決められた処理を行う。命令用キャッシュメモリ101の出力信号114、115はそれぞれ2命令分のビット幅を持っており、それゆえに命令読み出しは2サイクルに一度行えばよい。出力信号114は制御ブロック103でデコードされ、次のステージの信号118となる。この信号によって更に次のステージにおいてデータ120がそのまた次のステージにおいてデータ124が制御される。このように1つの命令を一連の処理に分割することによって命令の並列処理化を図るパイプラインを構成する。右半分を動作させる信号線115、119、121、126に関し

でも同様であるが、ここでは動作していないものとする。

これらの信号、即ち第1図の右半分を動作させるために、このプロセッサは特別な命令あるいは命令コード内に埋め込まれた特別なフィールドを有する。このような命令をここでは分裂命令 (Fusion Branch Instruction) と呼ぶことにする。この分裂命令がタイムスロット S1 に実行ユニット105で実行されたとすると、第2図における制御信号201がオンとなり、命令フェッチユニット102が動作可能となる。ユニット102は分裂命令によってセットされるアドレス、あるいは既にセットされているアドレスに従い命令用キャッシュメモリ101をアクセスする。これにより第1図の右半分のパイプラインも動作開始することになる。第1図の右半分と左半分は独立に動作する。

一方、このようにして独立に動作し始めたパイプラインを再び一本に絞り込む必要もある。そのため、分裂の場合と同様にこのプロセッサは特別な命令あるいは命令コード内に埋め込まれた特別

なフィールドを有する。このような命令を融合命令 (Fusion Branch Instruction) と呼ぶことにする。この融合命令がタイムスロット S2 に実行ユニット105で実行されたとすると、第2図における制御信号202がオンとなり、命令フェッチユニット100が動作を停止する。これにともない制御ブロック103及び実行ユニット105も停止され、左半分のパイプラインが完全に停止する。即ちパイプラインが右半分のみに絞り込まれる。融合命令によって停止したパイプラインは、動作中のパイプラインによる分裂命令あるいは一定時間ごとに行われるタイマチェック機構によってのみ再開される。また、この融合命令によって並列処理が停止されているとき、従来のスーパスカラ方式により停止された処理系を利用するようにすることもできる。

次に、このようなプロセッサを用いた場合の効果的なプログラミングについて第3図及び第4図を用いて説明する。ここでは、比較の対象として連続アドレスの2命令を同時に処理するスーパス

カラムシンをとりあげる。スーパスカラムシンはハードウェア物量が制約された条件、たとえば集積度が低いLSIを用いて実現する場合などに有効な方式であるが、連続アドレスの2命令という制約条件があるために並列度が向上しないという問題点がある。この方式による高速化の割合は、せいぜい1から2割程度と推定される。

第3図では実行サイクル数の異なる8命令(OP301~OP308)を両者で実行した場合を示している。簡単のために各命令の間にデータの競合関係はないものとする。OP301, OP304, OP305, OP308は実行に2サイクル要する命令であり、残りの命令は1サイクルにて実行可能とする。第3図の左側はスーパスカラムシンの例を示している。スーパスカラムシンは連続アドレスの2命令を同時に処理するため、それらの命令は同時に実行が開始される。そのため同時実行される命令の処理時間が異なると、ところどころ空きが生じ、全体の実行にタイムスロット T1~T8の8サイクルを要する。一方、第3図右側に示した本発明

による複数パス並列処理方法においてはこのような制約がないため、タイムスロット T1~T6の6サイクルにて実行を終了する。

以上は分岐のない簡単な場合の比較であるが、より重要な相違点として、並列化の割合が本質的に異なることがあげられる。即ちスーパスカラムシンにおいては連続の2命令が必ずしも同時に実行可能と出来ないのも、最適化コンパイラを用いてコードを生成したとしても並列化不可能な部分は本質的になくなる。分岐命令などがそのよい例である。スーパスカラムシンにおいては分岐は一度に1つしか実行できず、しかも分岐の入りと次の分岐命令との間の少ないコードの範囲内でしか並列度を抽出できない。分岐命令は5命令に1度出現すると言われており、平均4命令の間で並列度を抽出することになる。第4図に分岐命令を含むコードの例を示す(OP401~OP408)。OP403及びOP408は条件分岐命令であり、それぞれOP401, OP404へ分岐する。OP403は7回分岐を行い1回は分岐しない。またOP408は

5回分岐を行い1回は分岐しない。従ってOP 401からOP 403は8回、OP 404からOP 408は6回実行される。条件分岐命令において分岐する場合にはその実行に4サイクルを要し、分岐しない場合及びその他の命令は1サイクルにて実行が終了するものとする。第4図(a)に示したパイプラインの本数が1本の従来型のアーキテクチャをとった場合(シングルモード)は、実行命令が54、分岐が12回あるため、全体で90サイクル要することになる。第4図(b)に示したスーパスカラモードの場合には、OP 401と402、OP 404と405およびOP 406と407がそれぞれ並列可能であり実行される命令対の数は34、分岐はシングルモードと同様に12回あるので、全体で70サイクルとなる。これに対し、第4図(c)に示した本発明による複数バス並列処理方法を用いれば2つのループを別々のパイプラインに割り当てることができるため、分岐命令も含めて並列化が可能となる。即ちOP 401~403をプロセッサの左半分のパイプラインに割り当て、残りを右半分に割り当てる。プロセッ

サの左半分のパイプラインの実行時間は命令数が24、分岐回数が7であるため、全部で45サイクルとなる。また、プロセッサの右半分のパイプラインの実行時間は命令数が30、分岐回数が5であるため、全部でやはり45サイクルとなる。従って全体の実行時間は45サイクルとなり、シングルモードの90サイクルと比較して2倍の並列度を抽出したことになる。無論、分岐分岐に伴うオーバーヘッドはあるが、ループ回数が大きくなればこの影響は十分無視できる程度になる。これが本発明がスーパスカラモードに対して本質的に優れている理由である。

さて、このような複数バス並列処理方法を用いた場合において、2つのバス即ち2つのパイプラインの間にデータの依存関係がある場合についても考慮しなければならない。何故ならば、このような状況が生じる可能性は極めて高く、これに対する対策なくしては複数バス並列処理方法の本質的な高性能性を導き出すことは不可能となるからである。第5図は複数バス並列処理方法によるあ

るプログラムの流れを示したものである。本プロセッサのパイプラインは分岐命令501によって2つに分かれるが、それぞれバスは互いに関連をもって動作する。即ち手続き504において第1番目のレジスタあるいはメモリに対して書き込み(W1)が行われ、それをもう一方のバスの手続き509で参照する(R1)。手続き509はさらに第2番目のレジスタあるいはメモリに対して書き込み(W2)を行い、それを手続き512で参照する(R2)。また、同時に手続き508では第3番目のレジスタあるいはメモリに対して書き込み(W3)を行い、それを手続き513で参照する(R3)。その後、融合命令によってパイプラインの一本化が行われる。第5図のプログラムの流れはデータの依存関係によって、その実行順序が規定される。即ち手続き504は手続き509より先に、手続き508は手続き513より先に、そして手続き509は手続き512より先にそれぞれ実行されなければならない。このような実行順序はハードウェアでは制御不可能であるが、このプログラムのコードを生成する

コンパイラは実行順序の情報を持っているため、コード中に実行順序制御のための命令を埋め込むことができる。この命令を条件休止命令(Increment and Conditional Pause)と呼ぶことにして、その動作について説明する。前述の説明の通り、手続き504は手続き509より早く実行されなければならない。そこで、条件休止命令506、507を用いて同期をとる。2つのバスにはそれぞれ同期用のフラグ502、503があって、これらは分岐命令501によってリセットされ、条件休止命令によってその値を増加させられる。条件休止命令507は手続き509の実行を手続き504の実行終了まで待たせる働きをするものである。手続き504の実行終了時には、条件休止命令506が実行されており、フラグ502は1となっている。条件休止命令507はフラグ503を1にしてフラグ502と比較する。このとき、条件休止命令507はフラグ502の内容を信号線515を用いて参照できる。もしもフラグ502が0の場合、即ちフラグ502がフラグ503と等しくない時には、条件休止命令507はパイプラインを停止



して条件休止命令506の実行終了を持つ。条件休止命令510と511に関しても同様であるが、手続き512及び513はそれぞれ手続き509及び508が終了していないと実行開始不能であるので、相手のフラグと値が等しくない場合にパイプラインを停止するという判定条件を持つ条件休止命令を使用する。このようにして誤動作なく目的のプログラムを並列に実行できる。

〔発明の効果〕

本発明によれば、命令取り出し部分を複数もつことにより、分岐命令による並列度の減少を解決し、分岐命令自身をも並列化できるので、問題プログラムから並列度を100%抽出することができる。また、複数の命令実行部の同期をとるための命令とハードウェアを採用することにより、誤動作なく命令の実行順序を制御できる。

4. 図面の簡単な説明

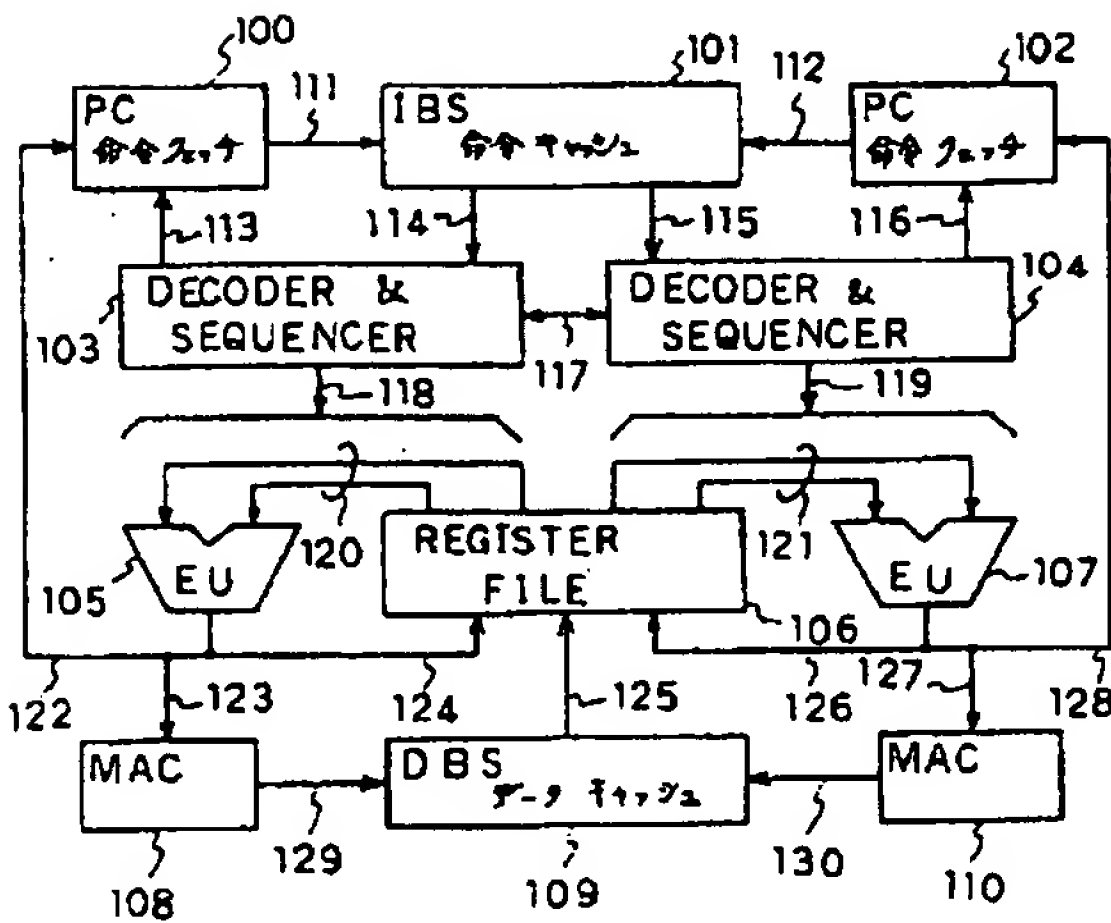
第1図は本発明の方法を適用した処理装置の一実施例を示すブロック図、第2図は第1図の処理装置におけるパイプライン制御の説明図、第3図

及び第4図は従来方法と本発明の方法によったときの処理例を示す図、第5図は命令間にデータ依存関係があるときの実行順序制御方法の説明図である。

100, 102…命令フェッチユニット、103, 104…制御ブロック、105, 107…実行ユニット、502, 503…フラグ。

代理人 弁理士 秋 本 正 実

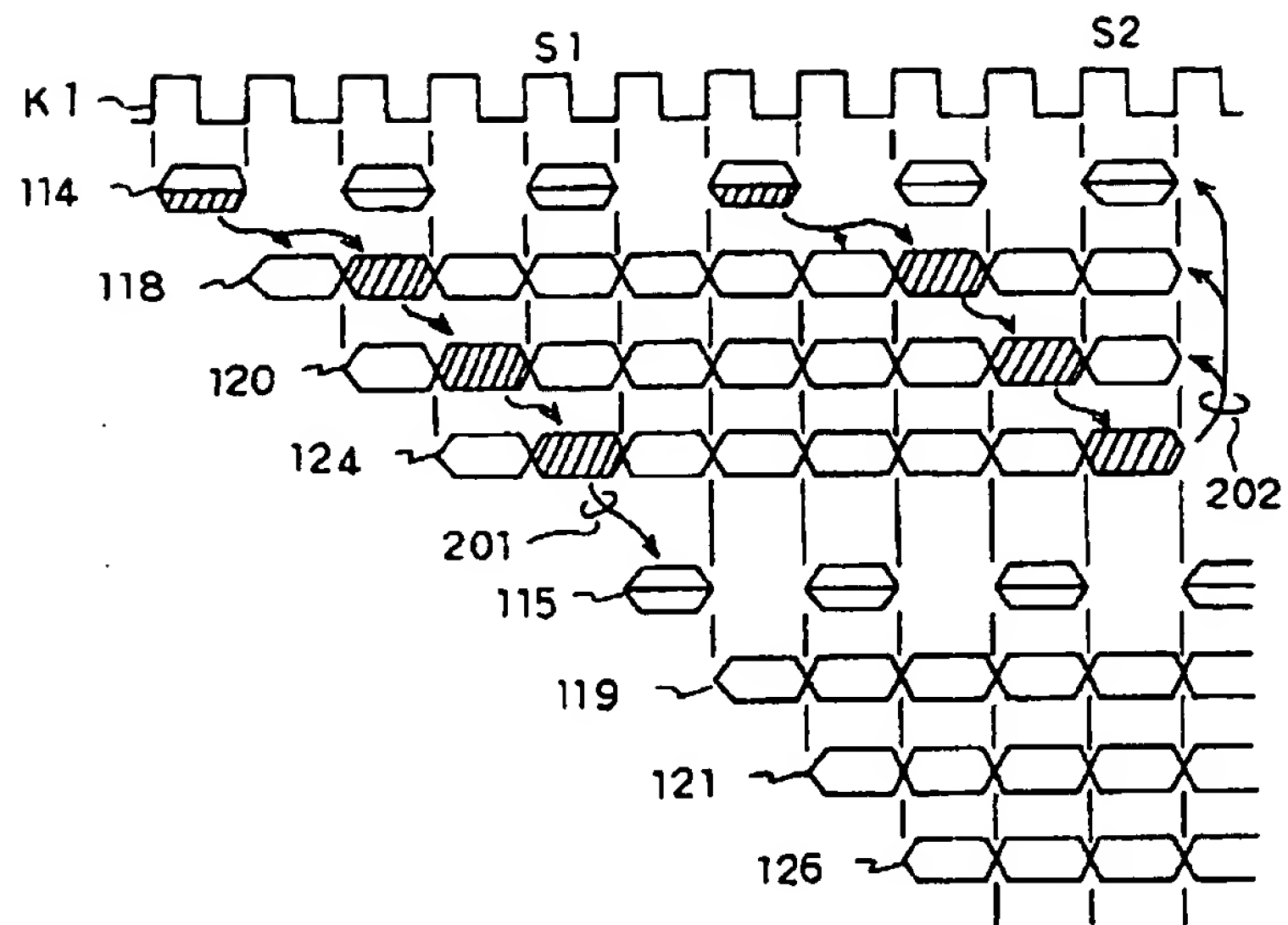
第 1 図



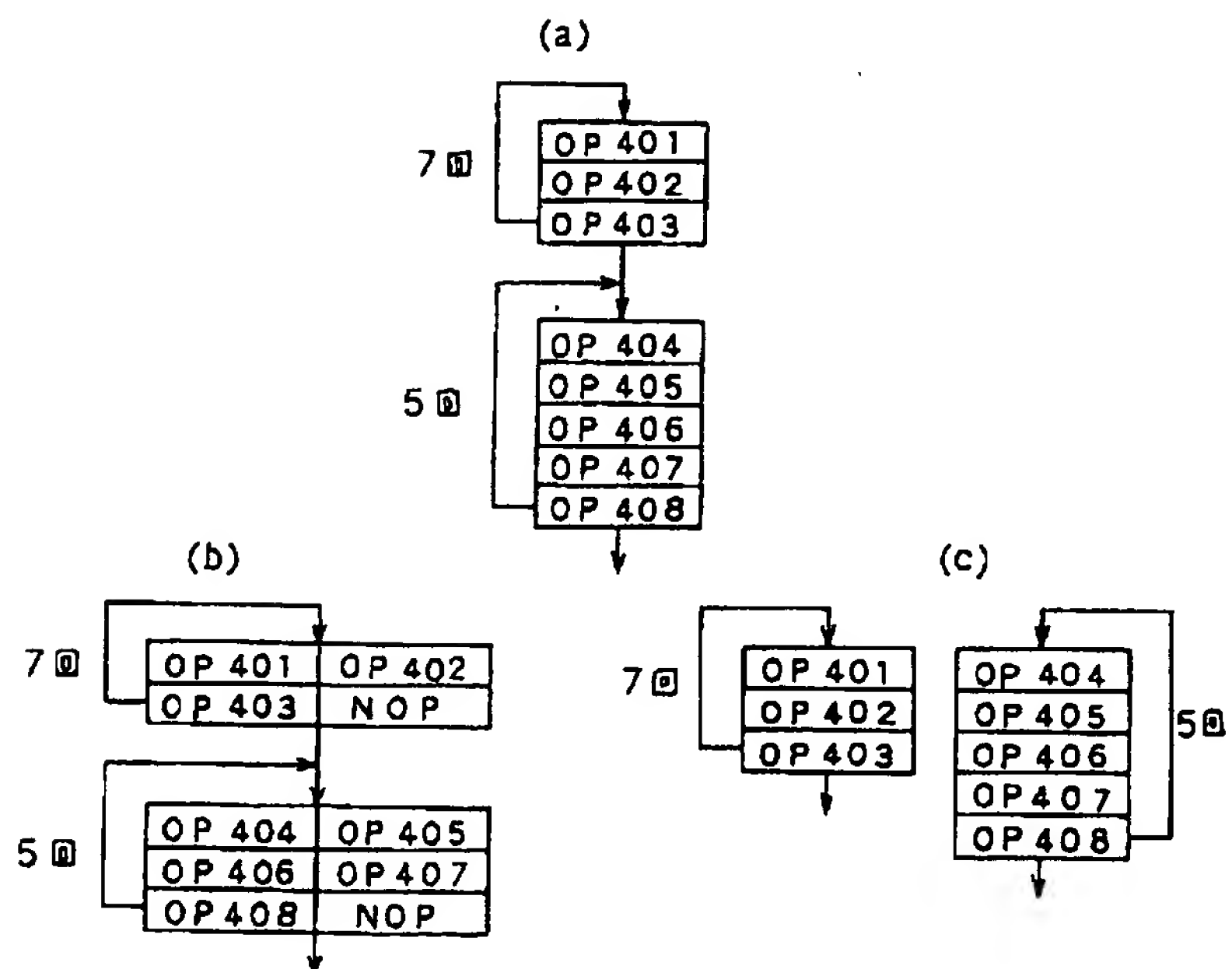
第 3 図

SUPER SCALAR		THIS INVENTION	
OP 301	OP 302	OP 301	OP 302
OP 303	OP 304	OP 303	OP 304
OP 305	OP 306	OP 305	OP 306
OP 307	OP 308	OP 307	OP 308

第 2 図

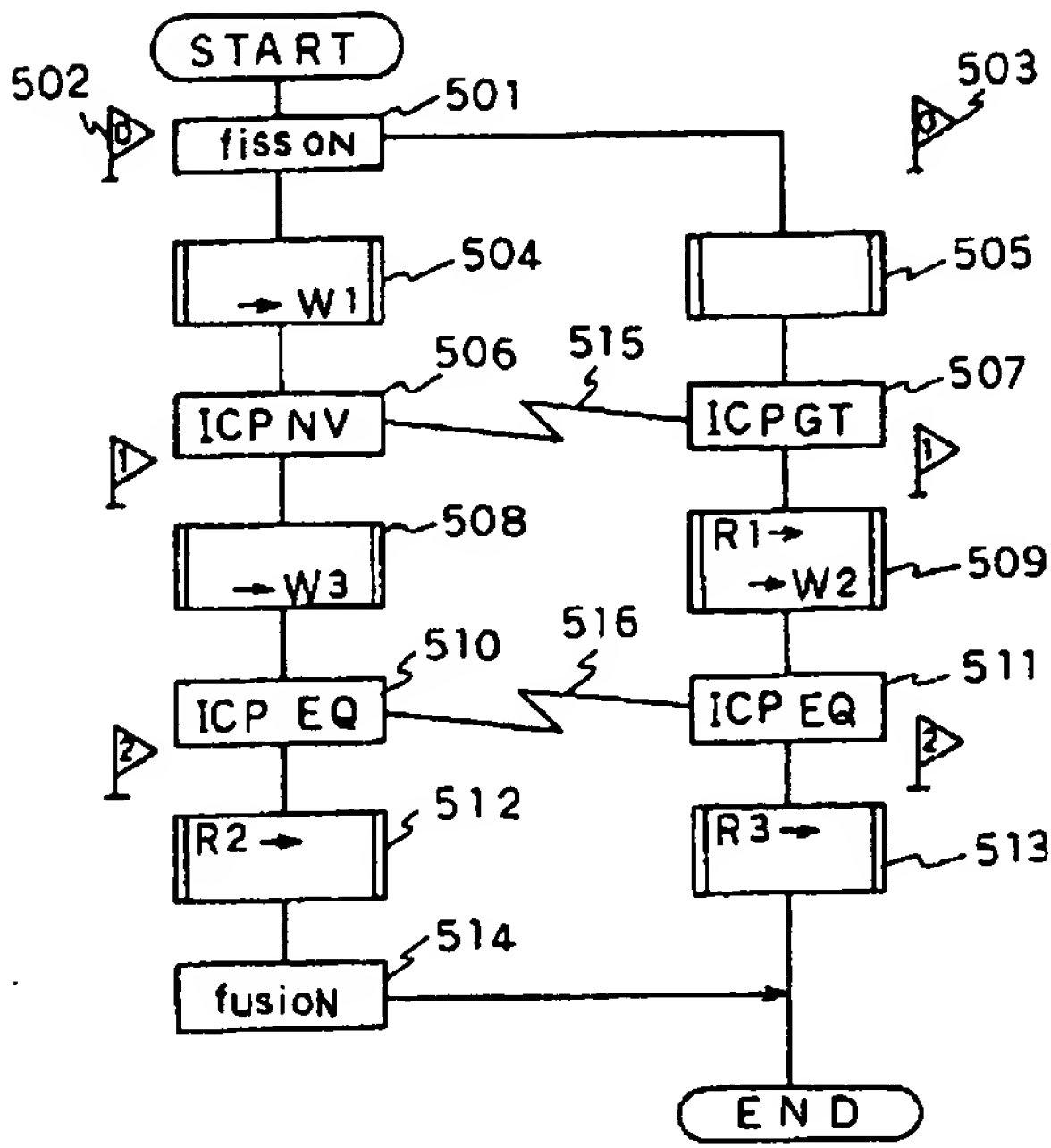


第 4 図





第 5 図



**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☒ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☒ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**